Attorney's Docket No.: 10559/430001 / P10444

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: ROAMING IN A COMMUNICATION NETWORK

APPLICANT: YLIAN SAINT-HILAIRE, JAMES L. JASON JR., FREDERICK WILLIAM STRAHM AND ERIK J. JOHNSON

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No EE647188565US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

March 19, 2001

Date of Deposit

Signature

Gil Vargas

Typed or Printed Name of Person Signing Certificate

ROAMING IN A COMMUNICATIONS NETWORK

BACKGROUND

This invention relates to roaming in a communications network.

Personal computers (PCs) can connect to networks including the Internet using various protocols including transmission control protocol/Internet Protocol (TCP/IP). When these PCs communicate over the Internet they are identified by an Internet Protocol (IP) address. Typically, the IP address does not change because the physical location of the PC is fixed. Mobile and portable wireless handheld devices such as a personal digital assistant (PDA) can connect to the Internet using wireless technology. However, these devices may experience interruptions in communications as they move, because the IP address associated with one location may be different from the IP address associated with another location.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a computer network system.

FIG. 2 is a block diagram of a mobile-device.

FIG. 3 is a block diagram of a home-agent.

FIG. 4 is a flow chart of a method of initializing a home-agent and mobile-devices.

FIGS. 5A-5D illustrate address parameters.

FIG. 6 is a flow chart of a method of processing requests from a mobile-device.

FIGS. 7A-7E represent a sequence of data units comprising a data packet transmitted from a mobile-device.

FIG. 8 is a flow chart of a method of generating a data-packet from a server.

FIGS. 9A-9D represent a sequence of data units.

FIG. 10 is a flow chart of a method of processing a change in a subnet by a mobile-device.

FIGS. 11A-11C represent address parameters during a change in a subnet.

DETAILED DESCRIPTION

As shown in FIG. 1, mobile-devices 20a, 20n exchange data with servers 26a, 26n using a computer network 28a, 28n through a home-agent 24. The computer network 28a, 28n can be, but is not limited to, the Internet, a local area network (LAN), or a wireless local area network (WLAN). The communications link 30, which can be implemented using wired or wireless technologies, is the connection point through which data flows over the network 28a, 28n. A mobile-device

20a, 20n includes a processor capable of connecting to the network 28a, 28n using wireless techniques. Each mobile-device 20a, 20n is assigned a real-address (RA) 21a, 21n by a dynamic host configuration protocol (DHCP) server 23 and a home-address (HA) 23a, 23n by the home-agent 24. The home-agent 24 is assigned a home-agent address (HAA) 25 and a media access control (MAC) address 27 that is based on a unique hardware number associated with the home-agent 24. The home-agent 24 is a network-compatible device that determines the network point to which data should be forwarded towards its destination.

The address relationship between the mobile-devices 20a, 20n and the home-agent 24 is based on a subnet-addressing scheme. The subnet may be a separate part of an organization's network and may represent all of the mobile-devices 20a, 20n at one geographic location, for example in a building or on a LAN. Dividing an organization's network 28a, 28n into subnets allows the mobile-devices 20a, 20n to be connected to the network with a single shared network address. By assigning each mobile-device 20a, 20n an address HA 21a, 21n based on the address HAA 25 of the home-agent 24, the home-agent 24 is able to acquire and maintain a single connection to the network for each of the mobile-devices 20a, 20n within the subnet.

Each server 26a, 26n is identified by a server-address (SA) 29a, 29n. The servers 26a, 26n communicate with the mobile-devices 20a, 20n over the network 28a, 28n through the communications link 30 using coaxial, optical, or wireless, or a combination of such techniques. Typically, a mobile-device 20a, 20n requests the services, such as data and information, provided by a server 26a, 26n through the use of the home-agent 24. As a mobile-device 20a, 20n moves from one location associated with an RA to another location associated with a different RA, uninterrupted communication is maintained between the mobile-device and the server.

As shown in FIG. 2, a mobile-device such as the device 20a can include a central processing unit (CPU) 32, such as an Intel Pentium-brand processor, connected to a computer-bus 38, such as a peripheral connect interface (PCI). The CPU 32 processes data and executes instructions residing in memory 34 such as a dynamic random access memory (DRAM). RA and HA addresses may be stored in memory 34. The mobile-device 20a also may contain an input/output (I/O) interface 36 coupled to the computer-bus 38. The I/O interface 36 allows peripheral devices 37 to be coupled to the mobile-device 20a. Peripherals 37 may include, for example, an input keyboard terminal, an output graphical display, and a mass storage device, such as a hard disk or a tape drive. A network

interface card (NIC) 54 provides the mobile-device 20a with access to the communications link 30. The technology supported by the NIC 54 can include, but is not limited to, wireless technologies such as those specified in IEEE 802.11, coaxial, or fiber.

The mobile-device 20a is driven and controlled by various levels of programs contained in software module 40. An operating system (OS) 42 is responsible for managing the overall functionality of the mobile-device 20a. Exemplary OSs include UnixWare, Windows NT, and Palm OS. Application programs 44, such as Web browsers or electronic-mail (Email) programs, are responsible for providing the user with a high-level interface to the mobile-device and the network. The application programs 44 are also responsible for providing the functions performed by the mobile-device 20a. A network protocol layer 46 is responsible for handling the reception and transmission of data packets over the network. Such a protocol may include a TCP/IP stack or other protocol based on the network open systems interconnections (OSI) communication model. The TCP/IP stack is responsible for managing the disassembly and addressing of data packets forwarded by the application programs 44 and the assembly of data packets received from a mobility-driver 48. Other protocol layers 46 may include AppleTalk which implements a datagram delivery

protocol corresponding closely to the network layer of the OSI communication model. The protocol layers 46 in the mobile-device 20a and the home-agent 24 should be similar for proper operation. The mobility-driver 48, which resides below the TCP/IP stack 46, is responsible for enabling the mobile-device 20a to move seamlessly between different locations. A user datagram protocol (UDP) layer 50 provides an additional standard communications protocol that is used in conjunction with the mobility-driver 48 to further process data to be exchanged over the network 28a, 28n. The network-driver 52 includes a program that controls the hardware-based NIC 54 enabling the mobile-device 20a to interface to the network 28a, 28n by facilitating the exchange of data over the communication link 30.

As shown in FIG. 3, a home-agent 24 includes some hardware and software components that are similar to those in the mobile-device 20a. For example, the home agent 24 includes a CPU 70 which processes data and executes instructions residing in memory 72. The CPU 70 and the memory 72 are both coupled to the device bus 73. The HAA and MAC addresses associated with the home-agent 24 are stored in the memory 72. The home-agent 24 maintains and keep tracks of the RA 70c and the HA 70d of each mobile-device 20a, 20n authorized to communicate with the home-agent. The NIC 68

provides the home-agent 24 with access to the communications link 30. The home-agent 24 also may contain an I/O interface 74 coupled to the device bus 73 which allows peripherals 75 to be connected to the home-agent. Peripherals 75 may include, for example, an input keyboard terminal, an output video display, and a mass storage device such as a hard disk or a tape drive.

The home-agent 24 is driven and controlled by various levels of programs contained in software module 56. The software module 56 includes an OS 58 responsible for managing the home-agent 24, application programs 60 responsible for providing the functions performed by the home-agent 24 such as managing the mobile-devices 20, 20n, and network protocol layer 61 such as a TCP/IP program stack for managing data packets. A proxy-driver 62 resides at a logically lower level then the TCP/IP layer 61 and is responsible for maintaining uninterrupted communication between the mobile-devices 20a, 20n and a particular server 26a, 26n. The proxy-driver 62 manages the data-requests, in the form of data-packets, generated by a mobile-device 20a, 20n and directed to a server 26a, 26n. It also manages the corresponding data-response from the server 26a, 26n in the form of data-packets. A network-driver 66 is responsible for controlling the NIC 68

and enabling the exchange of data-packets over the network 28a, 28n.

As shown in FIG. 4, to initialize the home-agent 24, it is assigned 80 a HAA according to a particular subnet addressing scheme. As shown in FIG. 5A, the HAA 92 includes a network-address-portion 92a which is set to 1.2, a subnet-portion 92b which is set to 3, and a host-address-portion 92c which is set to 4. The HAA 92 is an (IP) address and is represented, using dotted quad notation, as 1.2.3.4. In order for a mobile-device 20a, 20n to communicate with a server 26a, 26n over the network 28a, 28n, it contacts the home-agent 24 and requests 82 a HA. The home-agent 24 uses this addressing scheme to manage mobile-devices 20a, 20n that have a HA 94 (see FIG. 5B) including a network-portion 94a set to 1.2, a subnet-portion 94b set to 3, and a host-address-portion 94c set to at least 4. The home-agent 24 assigns, for example, a unique HA such as 1.2.3.5 to the mobile-device 20a. The subnet administrator is responsible for managing the allocation of these addresses. The assigned HA 94 is communicated 84 from the home-agent 24 to the mobile-device 20a and stored for later use.

Next the mobile-device 20a requests 86 a RA from the DHCP server 23 (see FIG. 1). The DHCP protocol allows network and subnet administrators to manage centrally and to automate the

assignment of IP addresses in an organization's network. The DHCP server 23 assigns a RA dynamically and avoids the necessity of the user entering a new RA every time the RA changes when the mobile-device 20a moves to a new location. The DHCP server 23 responds with a RA 96 (see FIG. 5C) including a network-portion 96a set to 200.300, a subnet-portion 96b set to 600, and a host-address-portion 96c set to 100. Once the mobile-device 20a, 20n receives 88 the unique RA from the DHCP server 23, it stores it in memory for later retrieval whenever it needs to communicate over the network 28a, 28n. The mobile-device 20a then communicates 90 the RA 96 to the home-agent using a standard roaming protocol "registration message" over the network 28a, 28n. The home-agent 24 then transmits 92 to the mobile-device 20a the MAC address 98 associated with the home-agent NIC 68. FIG. 5D shows a typical MAC address 98 associated with the home-agent 24 hardware. Once the mobile-device 20a has registered with the home-agent 24, it is now capable of communicating with a server 26a, 26n over the network 28a, 28n through the home-agent.

As shown in FIG. 6, a person using a mobile-device 20a, and desiring to communicate over a network 28a, 28n, such as the World Wide Web (WWW), can use an application program, such as a Web browser, to initiate 100 a request directed to a Web

site. As an example, a person using the mobile-device 20a and desiring to receive news information from a particular Web site would provide the address of the home page of the Web site. The Web browser application program provides the user with an interface to the Web. At the highest level of abstraction, a data-request includes a series of data-units that are combined into data-packets and communicated over the network 28a, 28n, processed by the home agent 24, and retransmitted to the server 26a. As shown in FIG. 7A, application-data-segment 500 is the first data-unit that is generated by the mobile-device 20a and it includes the data and information for the specific request.

Once the application-data-segment 500 has been generated, it is forwarded to the TCP/IP stack 46 of the mobile-device 20a which further processes 102 the data packet. The TCP/IP stack is a two-layer program in which the TCP portion is responsible for disassembling and assembling a data packet and the IP portion handles the address part of the packet so that it arrives at the correct destination. Alternatively, a UDP program layer can be used instead of TCP, and, together with IP, can generate a data-unit called a datagram. As shown in FIG. 7B, a TCP/IP-header 502 includes a TCP/UDP sub-header 502c and IP sub-headers 502a, 502b are concatenated to header 500. In a TCP embodiment, the header 502c provides

information associated with the packets that have been disassembled for transmission in order for the destination end to be able to reassemble the received packets. On the other hand, in a UDP embodiment, sequencing of the packets is not provided. Therefore, the application program at the receiving destination is responsible for assuring that the data packets arrive in the correct order. UDP can be used when ordering is not an issue or when the data units exchanged are small allowing network applications to save processing time. The IP program is responsible for setting the source-address field 502a to the HA of the mobile-device 20a and setting the destination-address-field 502b to the server-address SA of the server 26a requested by the mobile-device 20a.

After the TCP/IP-header 502 has been formed, the mobility-driver 48 handles the concatenation 104 of the link-layer-header 504 to the current data packet including headers 500 and 502. As shown in FIG. 7C, the link-layer-header 504 includes the MAC address field 504a which is set to the address corresponding to the unique hardware number of the home-agent 24. The MAC address is used by the MAC sublayer of the data-link layer (DLC) of the OSI model. The mobility-driver 48 then encapsulates 106 the data packet including headers 500, 502, 504 with an additional roaming-header 506 that includes an IP and a UDP portion. As shown in FIG. 7D,

the UDP portion 506c is set to the address of the proxy-driver 62 program residing in the home-agent 24. The IP portion includes the source-address-field 506a, which is set to the RA of the mobile-device 20a, and the destination-address-field 506b, which is set to the HAA of the home-agent 24.

Once the data-packet has been formed with the various headers 500-506, it is ready to be transmitted to the home-agent 24 over the network 28a, 28n. The data-packet is handed to the network-driver 52 in the mobile-device 20a corresponding to the physical layer of the OSI model. The network-driver 52 ensures that the data-packet is transmitted 108 over the network 28a, 28n.

After the data-packet is transmitted over the network 28a, 28n, it is received 110 by the home-agent 24 and handled by the NIC 68 in conjunction with the network driver 66. The data-packet is then forwarded to the proxy-driver 62, which is responsible for processing 112 the data-packet headers. As shown in FIG. 7E, the proxy-driver 62 removes the roaming-header 506 so that the data packet 508 includes the original headers 500, 502, and 504. The data-packet then is retransmitted 114 over the network 28a, 28n and directed to the server 26a associated with the destination address embedded in the destination-field 502b of the data-packet.

As shown in FIG. 8, the server 26a corresponding to the destination field 502b receives 200 the data-packet from the home-agent 24. The server 26a handles the data-packet using a network adapter and a corresponding network device driver. Application programs running on the server process 202 the request based on the information in the data packet. As shown in FIG. 9A, based on the earlier request from the mobile-device 20a, the server application responds with an application-data-segment header 600 containing news-related information in a format compatible with standard network communications protocols based on the OSI model. The server not only responds with the actual data, but it also adds information related to the source and destination of the data packet. As shown in FIG. 9B, the layers in the server 26a generate a response-header 602 containing a source-address-field 602a set to the IP address of the server and a destination-address-field 602b set to the HA of the mobile-device 20a. Once the data packet has been constructed with the communication headers, the server 26a transmits 204 the data-packet over the network 28a, 28n using, for example, standard network communications techniques.

Once the data-packet arrives 206 at the home-agent 24, the NIC 68 and the associated network driver 66 handle the data-packet. The data-packet is handed 208 to the proxy-

driver 62 which is responsible for processing and forwarding the data-packet to the appropriate mobile-device 20a, 20n. As shown in FIG. 9C, the proxy-driver 62 adds a roaming-header 604 with IP and UDP sub-headers 604a, 604b, 604c to the received data-packet headers 600 and 602. The data-packet subsequently is forwarded to the mobile-device 20a associated with the RA embedded in the data-packet. The proxy-driver 62 sets the destination-address field 604b to the RA of the mobile-device 20a and sets the source-address field to the HAA of the home-agent 24. Once the proxy-driver 62 processes the data packet, it is passed back down to the network-driver 66 which is responsible for transmitting 210 the data-packet over the network 28a, 28n.

The mobile-device 20a associated with the RA in the data-packet receives 212 the data-packet representing the response to the original server request for news related services. The various protocol layers 46 and the mobility-driver 48 of the mobile-device 20a process 214 the data packet according to their respective responsibilities so that the user request is satisfied. For example, the user may expect that the news information requested from the news service Web site will be received and processed by the Web browser running on the mobile-device 20a without an interruption in communication. As shown in FIG. 9D, the application-data-segment 606, which

is the data transmitted 600 from the server 26a in response to the mobile-device 26a, is available for processing by the application program 44 and the Web browser.

FIG. 10 illustrates a specific example of how uninterrupted communication between a mobile-device 20a and the server 26a is maintained when the mobile-device 20a moves to a location associated with a different subnet. The mobile-device 20 initially is assigned 300 a set of addresses when it registers with the home-agent 24. For example, as shown in FIG. 11A, the mobile-device 20a is assigned an initial-real-address RA1 700 and a home-address of HA 701. As shown in FIG. 11B, the home-agent 24 is assigned a home-address of HAA 702. The home-agent 24 keeps track of the mobile-device 20a by storing RA1 in its memory 72. The mobility-driver 48 of the mobile-device 20a uses these initial addresses to communicate 301 with the home-agent 24 over the network 28a, 28n.

The mobility-driver 48 is able to detect a change 302 in the real address corresponding to a different subnet. In a particular embodiment, the mobile-device is in communication with cell towers that emit signals of different magnitudes. As the mobile-device moves from a first location covered by a first cell tower to a second location covered by a second cell tower, the signal it receives from the first location will

decrease in strength while the signal from the second location will increase in strength. Once the strength of the signal from the second location reaches a threshold, the mobile device detects the change in location.

Once a change is detected, the mobile-device 20a requests 304 a new real-address from the DHCP server 23. As shown in FIG. 11C, the DHCP server 23 assigns a new real-address (RA2) 704 to the mobile-device 20a. Upon receipt of RA2, the mobile-device 20a communicates 306 the change in the real-address associated with the new subnet to the home-agent 24. The mobile-device 20a continues to communicate 301 with the home-agent 24 without a significant gap in communication as a result of the change in the subnet address.

The foregoing techniques can enable a mobile-device to maintain substantially uninterrupted communication while moving to another location associated with a new subnet address. Furthermore, since the technique can be implemented in a software layer that resides below the more complicated TCP/IP stack 46, there is no need to modify the TCP/IP stack, thus leading to a simpler and more cost-effective solution.

Various modifications may be made. For example, a mobile-device can communicate with another mobile-device within the same subnet by using a protocol such as internetwork packet exchange/sequenced packet exchange

(IPX/SPX). Various features of the system can be implemented in hardware, software, or a combination of hardware and software. For example, some aspects of the system can be implemented in computer programs executing on programmable computers. Each program can be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. Furthermore, each such computer program can be stored on a storage medium, such as read-only-memory (ROM) readable by a general or special purpose programmable computer, for configuring and operating the computer when the storage medium is read by the computer to perform the functions described above.

Other implementations are within the scope of the following claims.